VŠB - Technická univerzita Ostrava
Fakulta elektrotechniky a informatiky
Katedra informatiky

Uživatelsky rozšiřitelný slovník
User-Extensible Dictionary

2014 Vojtěch Hložánka

VŠB - Technická univerzita Ostrava
Fakulta elektrotechniky a informatiky
Katedra informatiky

Zadání bakalářské práce

Student: **Vojtěch Hložánka**

Studijní program: B2647 Informační a komunikační technologie

Studijní obor: 2612R025 Informatika a výpočetní technika

Téma: Uživatelsky rozšiřitelný slovník
User-Extensible Dictionary

Zásady pro vypracování:

Profesionální slovníky v elektronické podobě jsou běžně komerčně dostupné. Současným možnostem webu 2.0 odpovídá aktivní zapojení uživatelů obsahu a sdílení výsledků jejich spojeného úsilí pod vhodnou licencí. Tento trend může být vhodný pro budoucí kolaborativní rozšíření lokálních uživatelsky vytvářených slovníků, pokud budou komunitou přijaty a rozvíjeny. Dobře navržené a rozšiřitelné XML tagy, struktury a XML slovníková data, mohou být dobrým východiskem pro budoucí webový slovník.

1. Prostudujte současný stav volně dostupných slovníků a podobných iniciativ, zejména s důrazem na jejich datové struktury a propojení.
2. Navrhněte XML tagy a struktury vhodné pro tvorbu slovníku podporujícího uživatelskou rozšiřitelnost.
3. Navrhněte způsob tvorby, aktualizace a výměny/sdílení obsahu slovníku, případně řešení konfliktů nad obsahem.
4. Navrhněte uživatelské a aplikační rozhraní slovníku.
5. Implementujte funkční prototyp, analyzujte jeho přednosti a nedostatky a navrhněte budoucí rozšíření.

Seznam doporučené odborné literatury:

[1] Minder, P. - Bernstein, A.: How to Translate a Book Within an Hour, Towards General Purpose Programmable Human Computers with CrowdLang. WebSci 2012, June 22–24, 2012, Evanston, Illinois, USA. pp. 311-314, copyright 2012 ACM 978-1-4503-1228-8
[2] Navarro, A. - White, C. - Burman, L.: Mastering XML. Sybex
[3] Mlýnková, I. - Nečaský, M. - Pokorný, J. - Richta, K. - Toman, K. - Toman, V.: Technologie XML - Principy a aplikace v praxi. Grada 2008. ISBN 978-80-247-2725-7
[4] Virius, M.: C# 2010 Hotová řešení. Computer Press. 424 stran, ISBN: 978-80-251-3730-7

Formální náležitosti a rozsah bakalářské práce stanoví pokyny pro vypracování zveřejněné na webových stránkách fakulty.

Vedoucí bakalářské práce: **doc. RNDr. Petr Šaloun, Ph.D.**

Datum zadání: 01.09.2013

Datum odevzdání: 07.05.2014

doc. Dr. Ing. Eduard Sojka
vedoucí katedry

prof. RNDr. Václav Snášel, CSc.
děkan fakulty

Prohlašuji, že jsem tuto bakalářskou práci vypracoval samostatně. Uvedl jsem všechny literární prameny a publikace, ze kterých jsem čerpal.

V Hlučíně 30. 4. 2014 .....

Děkuji vedoucímu práce doc. RNDr. Petru Šalounovi Ph.D., že mi umožnil vypracovat práci, se kterou jsem se ztotožnil, a rovněž za jeho cenné rady a usměrnění projektu do výsledné podoby.

Abstrakt

Práce uvádí přehled současných řešení implementace slovníků, kde je důraz kladen na XML formát pro ukládání slovníku. Program slovníku byl vytvořen v jazyce C#. Slovník nabízí uživateli standardní uživatelské rozhraní s možnostmi tvorby vlastních překladů a jejich sdílení. Formát XDXF v XML, způsob uložení dat slovníku, umožňuje získání či aktualizaci obsahu slovníkových dat. Nabízí také perspektivní rozšíření možnosti práce jednotlivého uživatele i komunity uživatelů při dalším rozšiřování rozsahu a možnosti slovníku, včetně možného ovlivnění kvality jeho obsahu.

Klíčová slova: Překladový slovník, XDXF, XML, C#, .NET

Abstract

Dissertation presents an overview of current dictionary implementations, where the emphasis is on the XML format for storing dictionary data. The dictionary program was created in C#. Dictionary offers the user a standard user interface with ability to create custom translations and their sharing. Dictionary is saved in XML with XDXF formatting, which offers to obtain or update the content of dictionary data. It also offers promising options for work of single user or user community in the further expansion of the scope and possibilities of the dictionary, including the possible effect on the quality of its content.

Keywords: Translation dictionary, XDXF, XML, C#, .NET

Seznam použitých zkratek a symbolů

CAPTCHA	completely automated public Turing test to tell computers and humans apart
CRUD	Create Read Update Delete; vytvoření, čtení, editace a smazání záznamu
crowdsourcing	komunitní spolupráce
DOM	Document Object Model; objektový model dokumentu
GUI	Graphic User Interface; uživatelské rozhraní
OCR	Optical Character Recognition; optické rozpoznání symbolů
SP	slovníkový program
tag	XML značka/tag, významově je uzlem XML stromu
XDXF	XML Dictionary Exchange Format
XML	Extensible Markup Language; značkovací jazyk
XSD	XML Schema Definition; XML schéma
ZK	zdrojový kód

Obsah

1. Úvod

Profesionální slovníky v elektronické podobě jsou běžně komerčně dostupné. Současné možnosti webu 2.0 umožnují aktivní zapojení uživatelů. To v sobě zahrnuje příležitost spojit vzájemně své úsilí jak po stránce obsahu, tak i sdílení výsledků, a to pod vhodnou licencí. V budoucnosti může být tento trend vhodný pro rozšíření slovníků, které jsou uživateli vytvářeny společně - budou-li komunitou přijaty a rozvíjeny. Dobrým východiskem pro budoucí webový slovník mohou být vhodně navržené tagy, spolu se strukturami a slovníkovými daty v XML, které je možno dále rozšiřovat.

Slovník by bylo možné také naprogramovat jako webovou aplikaci s daty slovníku uloženými v databázi SQL, ovšem takových aplikací je na trhu velké množství. Proto jsem si jako cíl práce určil vytvoření funkčního prototypu programu, do kterého bude moci uživatel načíst slovníky dva, a to oficiální a uživatelský. V případě slovníku oficiálního může uživatel pouze vyhledávat klíčová slova (read only). V slovníku uživatelském má pak kromě toho možnost provádět i operace CRUD.

V textové časti, v 2. kapitole, jsou popsány slovníky - především slovníky volně šiřitelné. Následně, v 3. kapitole, jsou představeny použité tagy XML a jejich struktura, která umožnuje podporovat uživatele při případném dalším rozšiřování slovníků. Za tímto účelem byl vybrán již existující formát slovníku XDXF. Pak, v 4. kapitole, je navržena skladba slovníků, to jest jejich tvorba, aktualizace, výměna/sdílení a řešení konfliktů v obsahu. Kapitola 5. vysvětluje vzhled uživatelského rozhraní. Kapitola 6. popisuje aplikační rozhraní programu. Postupně jsou vysvětleny veškeré funkce programu s jeho přednostmi i nedostatky. Na závěr, v 7. kapitole, je prezentován návrh budoucího rozšíření programu (je napsán v jazyku C#[2]). Při vytváření tohoto programu posloužila jako užitečný zdroj informací stránka StackOverflow[1]. Manuál slovníku, který jsem navrhl, naleznete v příloze.

[1] Stackoverflow. *C#. Stackoverflow* [online]. 2013 [cit. 2014-03-31]. Dostupné z: http://stackoverflow.com/questions/tagged/c%23

2. Analýza dostupných slovníků

V současnosti je k dispozici řada různých slovníků a to jak překladových, tak výkladových. Tyto slovníky, ať už se jedná o komerční či ne, nejsou uživatelsky rozšiřitelné. U běžně dostupných slovníku uživatel nemůže přidat ke slovu další překlad, natož aby mu bylo umožněno přidat nové slovo, a to včetně překladu. Jsou pouze pro čtení.

2.1. Crowdsourcing

Lze přeložit jako komunitní spolupráce. Je to termín, který označuje využití komunity osob. Například k vyřešení problému, nebo tvorby jakéhokoli obsahu. Běžným příkladem je wikipedie[1].

2.1.1. CAPTCHA

Je to zkratka, která znamená plně automatický veřejný Turingův test k odlišení počítačů a lidí. Zobrazuje zdeformovaný text, který člověk dokáže přečíst, ale program ne.

reCaptcha

ReCaptcha[2](Obrázek 2) používá zapojení komunity uživatelů a aplikuje crowdsourcing k překládání knížek nebo jakéhokoliv textu, který nelze převést korektně z papírového formátu do digitálního pomocí OCR (Optical Character Recognition).

Obrázek 1 příklad reCapscha [5]

Obrázek 2 logo reCapscha [5]

Tato funkce využívá dvě skupiny znaků. Znaky, pro které systém zná správný „překlad“ a znaky, pro které správný překlad nezná (Obrázek 1). Každé slovo, které není možno přečíst korektně OCR je

[1] *Wikipedie* [online]. 2001 [cit. 2014-04-28]. Dostupné z: http://cs.wikipedia.org/wiki/Hlavn%C3%AD_strana
[2] What is reCAPTCHA. *ReCaptcha* [online]. 2014 [cit. 2014-03-31]. Dostupné z: http://www.google.com/recaptcha/learnmore

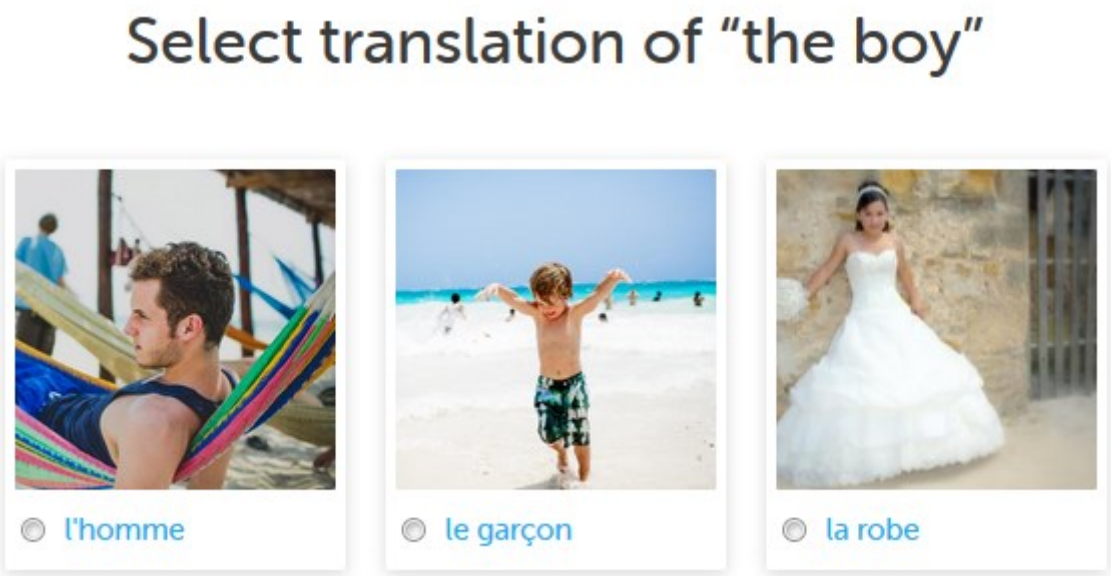
dáno uživateli zároveň se slovem, které je již známo. Uživatel je požádán, aby napsal obě tyto slova. Pokud napíše slovo, které je již známo správně, systém bude předpokládat, že je správně i druhé slovo. Systém toto opakuje u více uživatelů, aby se zvýšila pravděpodobnost správnosti odpovědi.

2.1.2. Duolingo[1]

Je systém, který umožnuje uživateli naučit se cizí jazyk a přitom zároveň překládat texty z internetu. Nabízí výuku italštiny, francouzštiny, španělštiny, němčiny a portugalštiny, ale předpokládá znalost angličtiny. Student prochází několika úrovněmi daného jazyka, přičemž jsou mu předkládány překlady, u kterých se předpokládá, že je zvládne. Pokud si však student není jist, že danou větu přeložil korektně, může si další překlady slov vyhledat na internetu. Systém také umožňuje uživateli hodnotit překlady jiných uživatelů. Tímto způsobem se student může dále poučit o tom, jaké jsou jiné možnosti překladu.

Dle oficiálního textu distributora obsaženého ve videu youtube [https://www.youtube.com/watch?v=WyzJ2Qq9Abs]:

> *Kdyby duolingo používalo 1 000 000 uživatelů, bylo by možné přeložit celou anglickou wikipedii do španělštiny za 80 hodin.*



Obrázek 3 příklad učení dle obrázků [6]

V lekci jsou k dispozici:

- Překlad slova, popřípadě věty.
- Napsání slova, nebo věty.
- Výběr správného překladu (Obrázek 3).
- Učení se novým slovům pomocí obrázku, popřípadě vyznačením v textu překladu.

[1] *Duolingo* [online]. 2013 [cit. 2014-03-31]. Dostupné z: https://www.duolingo.com/

2.2. Překladač Google

Je bezplatný automatický překladač, provozovaný od roku 2007 společností Google Inc. Služba překladu je zabudována v nejnovější verzi internetového prohlížeče Google Chrome, kde automaticky detekuje cizí jazyk a ve formě vysouvací nabídky pod adresním řádkem nabízí okamžité přeložení stránky či dokumentu. Čeština byla do překladače Google[1] zařazena v květnu 2008. V současnosti Překladač Google podporuje překlady mezi desítkami jazyků.

2.2.1. Použití

Tento překladač má mnoho funkcí, pro znázornění popíši pouze několik hlavních. Ostatní můžete nalézt v odkaze, který je poznámka pod čarou.

Okamžitý překlad textu

1. Navštivte stránku translate.google.com.
2. Vyberte jazyky pro překlad. Pokud si nejste jisti, v jakém jazyce je daná věta, klikněte na tlačítko **Rozpoznat jazyk**. S vyšším množstvím zadaného textu se přesnost automatického rozpoznání jazyka zvyšuje.
3. Začněte psát a překlad se začne ihned zobrazovat.

Čtení a poslech překladu

Pokud překládáte do jazyka, který nepoužívá latinku, zobrazí se u překladu tlačítko se symbolem **Ä**. Kliknutím na toto tlačítko se zobrazí překlad přepsaný latinkou.

U mnoha jazyků můžete také u přeloženého textu najít tlačítko s reproduktorem . Když na ně kliknete, uslyšíte vyslovenou verzi překladu.

Výsledky ze slovníku

Když přeložíte jedno slovo nebo běžnou frázi, může se pod překladem zobrazit jednoduchý slovník se slovními druhy a možnými alternativními překlady (Obrázek 4). Vedle každého slovníkového hesla uvidíte odpovídající sadu zpětných překladů do původního jazyka. Pruh vedle každého hesla ukazuje, jak běžný je daný překlad na internetu.

Psaní rukou

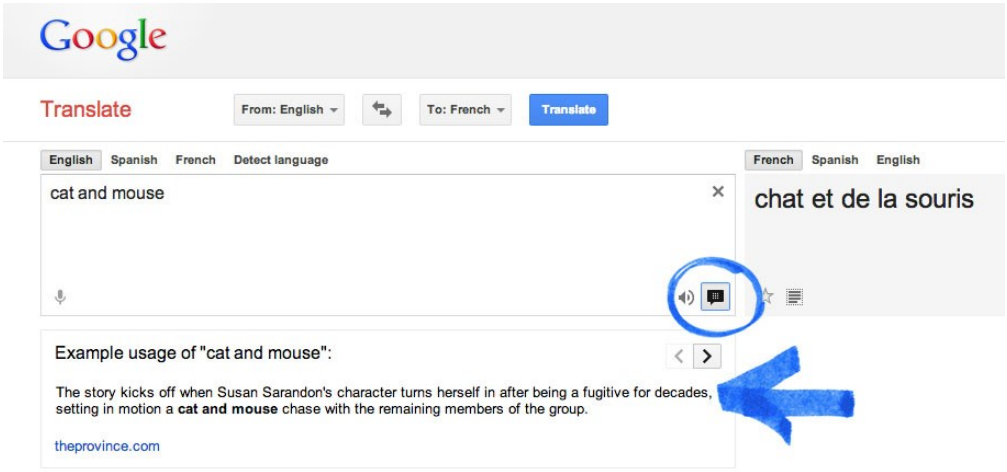
Nástroj pro psaní rukou umožňuje přeložit napsaný výraz, i když nevíte, jak znaky napsat na klávesnici. Místo toho znaky nakreslíte na obrazovce a ihned se zobrazí překlad. Nejprve zvolte jazyky pro překlad a pak pomocí ikony nástrojů zadávání přepněte na psaní rukou.

[1] GOOGLE. *Google translate* [online]. 2007 [cit. 2014-03-31]. Dostupné z: https://support.google.com/translate/#

Obrázek 4 Ukázka výsledků [4]

Zobrazení příkladů použití slov

Pokud hledáte překlady jednotlivých slov nebo frází (Obrázek 5), je často užitečné vidět překlad v kontextu. Kliknutím na ikonu příkladů použití u překladu zobrazíte příkladovou větu obsahující překlad.



Obrázek 5 Zobrazení příkladů [4]

Překlad celých webových stránek

Celou webovou stránku můžete přímo z Překladače Google přeložit tak, že do jeho zadávacího pole zadáte adresu stránky (např. `www.google.com`) a kliknete na „přeložit“. Pokud máte prohlížeč Chrome, tak můžete přeložit celou stránku pomocí vysouvací nabídky pod adresním řádkem. Je možné si takto nastavit automaticky překlad vždy, když je stránka v určitém jazyce.

Virtuální klávesnice

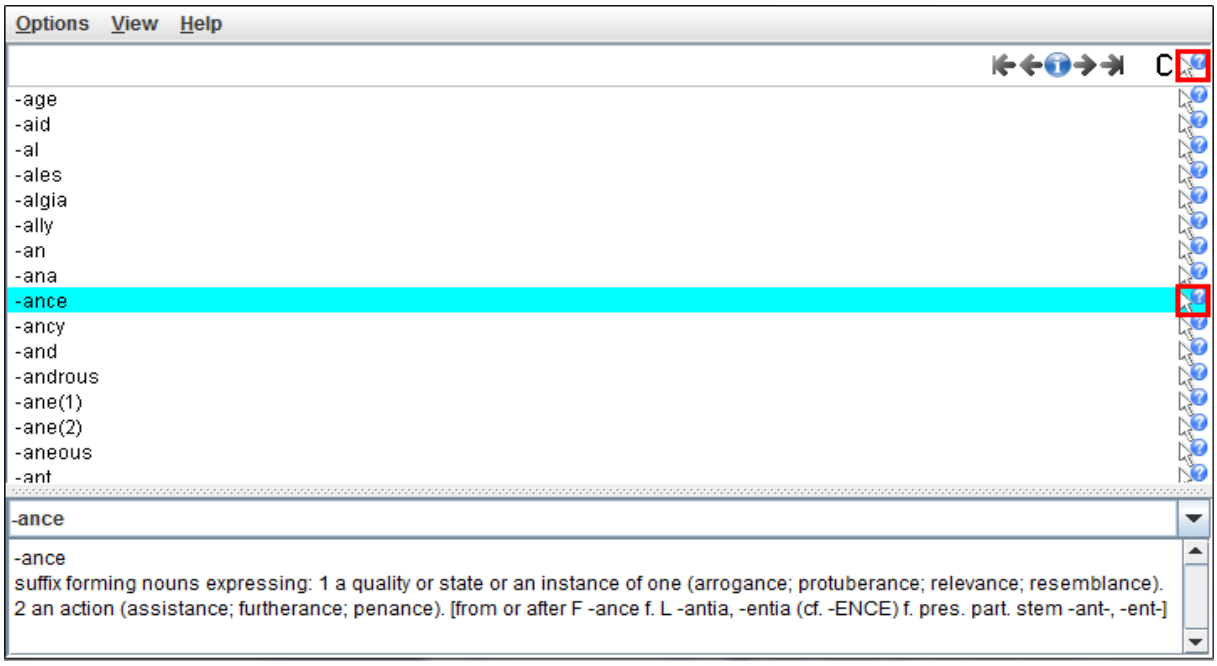
Nástroj zadávání v podobě virtuální klávesnice otevře na obrazovce klávesnici pro konkrétní jazyk. Tyto klávesnice jsou navrženy speciálně pro zadávání textu ve vybraném jazyce. Pokud máte vstupní jazyk nastaven například na ruštinu, můžete si zvolit klávesnici s azbukou. Virtuální klávesnici lze ovládat pomocí skutečné klávesnice nebo klikáním na klávesy virtuální klávesnice.

2.3. Atlantida

Je to open-source vícejazyčný slovník napsaný v Javě. Umí překládat slova z jednoho jazyka do druhého a vyslovit je. Od alfa verze 0.15, Atlantida[1] používá XDXF formát pro ukládání slovníku. Na tento slovník se vztahuje GNU GENERAL PUBLIC LICENSE[2].

2.3.1. Grafické uživatelské rozhraní

Grafické rozhraní tohoto slovníku je velice jednoduché (Obrázek 6). V první části máme slova, která hledáme a pod nimi je jejich překlad, synonyma, užití, popis a podobně. Posuvník („scroll bar“) zde u hledaných slov také chybí, což při vyhledávání slova může ztížit práci.



Obrázek 6 Ukázka grafické rozhraní Atlantida [3]

2.3.2. Funkcionalita

Slovník používá Javu staršího data, tudíž se mohou vyskytnout problémy s jeho kompatibilitou. Při spuštění programu se načte celý slovník z XDXF souboru (např. `eng-cz.xdxf`) do paměti. Během načítání je program velice náročný na paměť, ale po načtení slovníku náročnost klesá. Přesná čísla se mění dle velikosti slovníku. V případě, že programu dodáme audio soubory, můžeme si dané slovo přehrát. Používá Jazzy[3] (Java library for Spell Checking), kontrolující pravopis.

[1] *Atlantida Multilingual Dictionary* [online]. 2006 [cit. 2014-03-31]. Dostupné z: http://atla.revdanica.com/en/
[2] *GNU GENERAL PUBLIC LICENSE* [online]. 2007 [cit. 2014-03-31]. Dostupné z: http://www.gnu.org/copyleft/gpl.html
[3] *The Java Open Source Spell Checker* [online]. 2005 [cit. 2014-03-31]. Dostupné z: http://jazzy.sourceforge.net/

Po prvním načtení slovníku si program vytvoří několik souborů.

- `words.db` obsahuje seznam všech slov ve slovníku.
- `config.cfg` obsahuje konfiguraci (dole).

```
#Atlantida config file.
#Mon Mar 31 14:56:59 CEST 2014
cache_format_version=1
divider_location=174
splitPane_size_y=350
splitPane_size_x=893
dics_path=C\:\\Users\\User0\\Documents
```

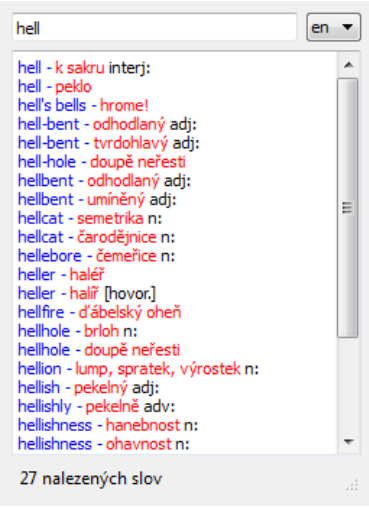
Zdrojový kód 1 příklad konfiguračního souboru [3]

- Každý slovník má vytvořeny soubory s příponou `.fork .hash .list`, tyto soubory mají jméno dle souboru slovníku.
- `alphabet_regions a key_starts` tyto soubory nemají žádnou příponu.

2.4. SlovniQ

SlovniQ[1] je lehce ovladatelný a velice rychlý anglicko-český slovník. Pracuje se slovníkovými daty z GNU/FDL Anglicko-českého slovníku.

Cílem bylo vytvořit program, do kterého se pouze zadá překládané slovo. Nekoná se stahování/aktivace slovníku, různé nastavování, dokonce se ani nemusí vybírat jazyk. Program sám hledá v obou jazycích. Jediné co můžete ovlivnit je řazení výsledků. Je napsán v jazyce C++ a používá knihovnou Qt. Pokud slovo při vyhledávání má více překladů, je klíčové slovo vypsáno vícekrát (Obrázek 7).



Obrázek 7 GUI SlovnikQ [7]

[1] NĚMEC, Jiří. *SlovniQ* [online]. 2010 [cit. 2014-03-31]. Dostupné z: http://code.google.com/p/slovniq/

2.4.1. Vnitřní funkce

```
if (!this->dataFile.isOpen() )
{
    this->dataFile.setFileName(localFileName);
    if ( !this->dataFile.open (QIODevice::ReadOnly |QIODevice::Text) )
    {
        emit zmenaStatusu ( "Nejde cist z disku" );
    }
}
```

Zdrojový kód 2 kontrola čtení souboru [7]

V tomto příkladu kódu (nahoře) se zkontroluje, jestli je soubor otevřen a poté jestli lze z něho číst. Také si uloží jméno souboru do paměti.

```
void SlovnikData::stahnout()
{
    httpSoubor.setFileName(httpNazevSouboru);
    httpSoubor.flush();
    if ( !httpSoubor.open ( QIODevice::WriteOnly ) ) {
        emit zmenaStatusu (QString::fromUtf8( "Nejde zapisovat na disk"
));
    }
    http.setHost ( "slovnik.zcu.cz");
    emit zmenaStatusu ( QString::fromUtf8("Stahuji slovník." ));
    httpGetId = http.get( "/files/slovnik_data_utf8.txt.gz" ,&httpSoubor
);
    connect ( &http, SIGNAL ( done ( bool ) ),this, SLOT (
stahovaniUkonceno ( bool ) ) );
}
```

Zdrojový kód 3 Metoda stáhnutí slovníku [7]

Program je schopen si stáhnout slovník, pokud ho už neobsahuje (nahoře). Tato funkce se obvykle neprovádí, protože program je distribuován s daty. Kontroluje schopnost zápisu na disk a metoda stahovaniUkonceno (bool) kontroluje, pokud nedošlo k chybě (dole).

```
void SlovnikData::stahovaniUkonceno ( bool error ) {
    if ( error )
        emit zmenaStatusu ( QString::fromUtf8("Doslo k problému při
stahování slovníku") );
    else {
        emit zmenaStatusu ( QString::fromUtf8("Slovník byl stažen.") );
        httpSoubor.close();
        .
        .
        .
        emit zmenaStatusu ( QString::fromUtf8("Slovník byl rozbalen.") );
        this->vytvorIndex(true);
    }
}
```

Zdrojový kód 4 Metoda stahování ukončeno [7]

2.4.2. GNU/FDL Anglicko-Český slovník[1]

Program používá data z tohoto slovníku a i já je používám pro základní slovník. Data ve slovníku jsou formátována jednořádkově, na jednom řádku je hledané slovo i překlad (Zdrojový kód 5). Obsahuje 87 485 různých překladů.

```
a blast        zaútočit - napadnout            v:     tata
a blast        zničit                          v:     tata
a blast        vyhodit do vzduchu              v:     tata
a blast        odstřelit pomocí trhaviny       v:     tata
```

Zdrojový kód 5 Příklad překladu [7]

Cituji autora Jiřího Němce ze zdroje [http://code.google.com/p/slovniq/]:

> *Rozhodl jsem se vytvořit svobodný anglicko-český slovník. Cílem je vytvořit dostatečně rozsáhlý slovník, který by bylo možno stáhnout na svůj počítač. Chci vytvořit tento slovník, aniž bych vykradl existující elektronický či papírový slovník. Tudíž mi nezbývá nic jiného než začít od nuly.*

Shrnutí

Provedl jsem analýzu, jak profesionální slovníky pracují a fungují. Těmito slovníky jsem se inspiroval, ať už po funkční, nebo grafické stránce a poznatky uplatnil. Velmi inspirativní také pro mě bylo zjištění, jak velký kreativní potenciál v sobě má komunita osob, která tímto způsobem spolupracuje na překladu knih, nebo jejich převedení do digitálního formátu. Já tyto znalosti použiji k vytvoření slovníku. Rád bych v budoucnosti tyto znalosti použil k vytvoření slovníků pomocí crowdsoucingu.

[1] *GNU/FDL Anglicko-Český slovník* [online]. 2004 [cit. 2014-03-31]. Dostupné z: http://slovnik.zcu.cz

3. Struktura XML

XDXF je zkratka pro XML Dictionary Exchange Format[1], a jak jeho název napovídá, určuje formát dat slovníku. Některé další slovníky podporují XDXF (viz GoldenDict, XDClient).

Každý slovník je umístěn v své vlastní složce a název složky se používá jako ID. Musí obsahovat pouze znaky latinské abecedy a nesmí obsahovat mezery nebo jiné speciální znaky. Takže, pokud jméno slovníku je "`Webster's Unabridged Dictionary published in 1913`", pak název složky by mohl být např. "`Webster1913`". Soubor obsahující samotný slovník se pak musí vždy jmenovat "`dict.xdxf`". Je doporučeno, aby každý slovník měl sadu ikon pro panel nástrojů a velkou ikonu na hlavní straně. Rozměry by měly být: 16×16, 32×32, 512×512. A názvy souborů `icon16.png` `icon32.png` a `icon512.png`. Názvy souborů rozlišují malá a velká písmena.

Všechny XDXF slovníky mají příponu "`.xdxf`". Jsou ve formátu XML s jakýmkoliv kódováním Unicode (obvykle UTF-8). Jakékoliv jiné kódování je přísně zakázáno.

3.1. Tagy

```
<xdxf lang_from="XXX" lang_to="XXX" format="FORMAT" revision="DD">
```

XML 1 Formát hlavičky [8]

Kořenový element, musí mít 4 atributy (XML 1).

- `lang_from`: 3písmenné slovo, reprezentující z jakého jazyka je slovník.
- `lang_to`: 3písmenné slovo, reprezentující do jakého jazyka překládáme;
  oba atributy jsou dle normy ISO 639-3.
- `format`: Atribut definuje výchozí formátování slovníku a může být buď `visual`, nebo `logical`. Výchozí formát může být přepsán pro konkrétní položky (viz níže).
  - Ve formátu `visual` jsou položky formátovány vizuálně. Jsou určeny pro slovníkové programy (dále uvedený jako SP). Bez vkládání nebo odstraňování mezer nebo EOL (konec řádku). Nicméně, SP mohou označit obsah některých logických tagů (jako <gr> nebo <abbr>) s různými barvami.
    Vizuální formát se nedoporučuje! Formát XDXF je vyvinut speciálně pro logicky strukturované slovníky. Vizuální formát je zde poskytnut pouze za tím účelem, aby umožňil kompatibilitu se slovníky, které jsou převedený ze starých formátů (plain-text).
  - V logickém formátu položky formátovány vizuálně nejsou.
- `revision`: určuje verzi ve kterém je soubor XDXF naformátován.

[1] XDXF Description. *Github* [online]. 2013 [cit. 2014-04-01]. Dostupné z: https://github.com/soshial/xdxf_makedict/blob/master/format_standard/xdxf_description.md

Slovník je rozdělen do dvou částí: <meta_info> a <lexicon> (XML 2).

```
<xdxf ...>
    <meta_info>
        All meta information about the dictionary: its title, author etc.
    </meta_info>
    <lexicon>
        <ar>article 1</ar>
        <ar>article 2</ar>
        <ar>article 3</ar>
        <ar>article 4</ar>
        ...
    </lexicon>
</xdxf>
```

XML 2 rozložení dokumentu [8]

<meta_info> obsahuje veškeré „meta“ informace ohledně slovníku.

1. <title> Zkrácený název slovníku napsán v angličtině.
2. <full_title> Celý název slovníku. Může obsahovat ne-anglický název.
3. <publisher> Oficiální vydavatel slovníku (nepovinný údaj).
4. <authors> Obsahuje seznam <author> tagů a obsahuje všechny osoby (organizace), které se podílely na tomto slovníku (nepovinný údaj).
    - <author role="xxx"> Jeden tag pro každého autora.
      Jedena osoba může mít více rolí, tudíž by měla být uložena pomocí dvou <author> tagů
5. <description> Popis slovníku od vydavatele. Doporučuje se zahrnout následující: Autorská práva, licence, odkud tento soubor lze stáhnout, odkud neformátovaný soubor (tj. původní soubor slovníku před konverzí do formátu XDXF) může být stáhnut, odkud původní neformátovaný slovník soubor byl získán, a odkaz na skript, který byl použit k převodu původní neformátovaný soubor slovníku do XDXF. Popis může obsahovat XHTML tagy, které jsou povoleny v XDXF (viz níže).
6. <abbreviations> je seznam <abbr_def> tagů (XML 3). Obsahuje všechny zkratky používané pro popisky ve slovníku (gramatické popisky např.). <abbr_def> tag definuje zkratky a obsahuje dva různé tagy.
    - <abbr_k> Zkratka pro klíč zkratky: zkrácený text.
    - <abbr_v> Zkratka pro hodnotu: full text, popis zkratky, který se zobrazí, když nad ním podržíme kurzor.

    <abbr_def> Může obsahovat type atribut, který uvádí, jaký typ popisku tato zkratka je:

    - <abbr_def type="grm"> — Uvádí gramatické vlastnosti slova (podstatné jméno, příčestí minulé, atd.).
    - <abbr_def type="stl"> — Stylistické vlastnosti slova (vulgární, archaický, poetický, atd.).

- <abbr_def type="knl"> — Oblast/doména znalostí (počítače, literatura, gastronomie, typografie atd.).
- <abbr_def type="aux"> — Jednoduché pomocné slova jako ("např.", "tedy", "zřídka", atd.).
- <abbr_def type="oth"> — Ostatní.

```
<abbreviations>
<abbr_def type="grm"><abbr_k>n.</abbr_k><abbr_v>noun</abbr_v></abbr_def>
<abbr_def type="knl"><abbr_k>polit.</abbr_k><abbr_v>politics</abbr_v>
</abbr_def>
<abbr_def><abbr_k>Av.</abbr_k><abbr_k>Ave.</abbr_k><abbr_v>Avenue</abbr_v
>
</abbr_def>
</abbreviations>
```

XML 3 Příklad abbreviations [8]

7. <file_ver>, <creation_date> jsou povinné informace. <last_edited_date>, <dict_edition>, <publishing_date>, <dict_src_url> jsou nepovinné „meta“ informace. Všechna data by měla být formátován jako DD-MM-YYYY. Pokud datum není zcela známo, použijeme nuly: 05-00-2011.

<lexicon> obsahuje veškeré <ar> (položky) tagy. Skupiny <ar> tagů spojují dohromady věci týkající se jedné z klíčových-frází.

Mohou mít volitelný atribut f, např. <ar f="x">, který může mít hodnotu buď *v* (visual) nebo *l* (logic), a mohou být použity k potlačení výchozího formátování slovníku, které bylo uvedeno v <xdxf> tagu.

Následující dva tagy jsou povoleny pouze mezi <ar></ar> tagy.

I. <k> Klíčová fráze je jedinečný sled písmen/ideogramů, kterým je tag identifikován a může být nalezen. Může obsahovat více než jednu klíčovou frázi, ale vždy nejméně jednu. Pokud existuje více než jeden <k>, SP by měl zobrazit všechny varianty klíčových-frází. Tag <k> nesmí být vnořeny v jiném <k>.
    - <opt> Volitelná součástí klíčové-fráze. Položka je prohledána pomocí obsahu <k> bez obsahu <opt>, ale ukazuje se v položce s ním. Tag <opt> může být použit pouze mezi <k></k> tagy.

II. <def> Označuje celé tělo položky slova, definice, skupiny definic, které spadají do určité kategorie.
    Tip. Tyto kategorie mohou být různé části řeči, nebo mají jiný etymologie, například, nebo stejný smysl s různými konotacemi.
    <def> tagy se mohou vnořovat a obvykle dělají, pokud položka není překlad „1 slovo-na-1 slovo“.
    Tag <def> musí být uvnitř <ar>, i když položka je jednoduchá a není třeba vytvořit skupinu.
    * Mohou mít cmt (komentář) atribut, který pomáhá rozlišit jednu definici od ostatních.

* Mohou mít unikátní tribut `id` [01-9a-zA-Z], na který může být odkaz v jiné položce.
* Mohou mít integer/double atribut `freq` (frekvence): absolutní nebo relativní četnost hodnoty definice.

V položkách s tagy `visual` formátu `<def>` nemají vliv na formátování. Pro položky, které mají logický formát SP musí rozlišovat vizuálně jednu definici od druhé podle úrovně vnoření pomocí odsazení, zmenšení velikosti písma nebo číslováním definic '1) ', '2) "... nebo 1. ', '2. "... nebo "A.", "B." ... atd. před každou definicí.

1. `<gr>` Určuje gramatické informace o slově. Mohou obsahovat různé tvary slova, použití slova, gramatické popisky a další informace o tohoto druhu.
2. `<tr>` Označuje přepis/výslovnost informace, IPA symboly jsou výchozí. Může mít také "mode" atribut s hodnotami `X - SAMPA` nebo `erkIPA` .
3. `<kref>` Jednoduchý odkaz na jiný klíčovou frázi, která se nachází ve stejném souboru. Pro další informace viz `<sr>`.
4. `<dtrn>` Tento tag označuje přímý překlad klíčové fráze (obvykle nepoužívané pro vysvětlující slovníky). V ne-výkladových slovnících by měl být tag použit pro pomoc softwaru, aby slovník mohl automaticky extrahovat základní a nejjednodušší překlady klíčové fráze. Může to být užitečné pro:
    - Automatická extrakce dat pro nápovědné překlady (např. jako qDictionary).
    - Zviditelnit `<dtrn>` v seznamu slov, aby se zabránilo příliš častému vyhledávání „celé položky“.
      Tato slova by měla být vyznačena pro jednodušší prohledávání napříč položkami (obvykle tato slova jsou zobrazena jako tučné, někdy také v tmavě - oranžové). Tato slova mohou také automaticky vypadat jako odkazy (`<kref>`), které vedou k položce opačného jazykového páru.
5. `<rref>` Odkaz na zdroj audio souborů (`mp3`, `ogg` nebo `opus` formát), který by měl být umístěn ve stejné složce jako slovník (XML 4).
    - volitelné `start` a `size` atributy jsou nezbytné pro audio a video soubory, kdy referenční body ukazují na určité části souboru. To je velmi výhodné, abychom měli všechny audio data slovníku v jednom souboru, s odkazem na různé části z různých překladů. Atribut `start` určuje offset: pozici v souboru prvního bajtu bloku zájmu, a `size` určuje jeho délku v bajtech. Je-li `start` atribut vynechán, pak se předpokládá, že to je 0. Je-li atribut `size` vynechán, pak se předpokládá, že soubor má být přehrán až do konce.

```
<rref start="xxx" size="xxx">crawl</rref> nebo <rref
start="xxx" size="xxx"/>
```

XML 4 Použití `rref` [8]

6. `<iref href="http://www.somewebsite.com">` Odkaz na zdroj v Internetu. Zachování předponu („http://“, „https://“), je povinné.

7. <abbr> Označuje zkrácený tag. Tagy by měly mít vysvětlení uvedeny v korespondenčním <abbr_k> tagu v <abbreviations> části.
8. <c c="xxxxxx">...</c> (označuje 32 bitový (6-místný) hexadecimální kód barvy) Označuje text danou barvou. Syntaxe pro "c" atribut je stejný jako pro "color" Atribut "font" tagu v HTML. Pokud je atribut barva vynechán, bude vybraná výchozí barva. Výchozí barva je vybraná SP.
9. <ex> Označí text příkladu (obvykle uvedený v šedé nebo jiná barvě podle SP). Tip (indexování): příklady, musí být indexovány implicitně, ale uživatelé by měli být schopni nastavit, zda chtějí příklady, které budou indexovány a vyhledávány, nebo ne. Atribut type, může být:
    - exm - obvyklé příklady s nebo bez překladů
    - phr - může obsahovat jakýkoli typ frazémů (idiomy, kolokace, klišé atd.)
    - prv – přísloví
    - oth – jiné

    Atributy source a author určují, odkud byl příklad převzat.
    Měl obsahovat tagy: <ex_orig> pro původní věty příkladu (v hodnotě: 1 nebo více).
    <ex_tran> Je volitelný, může být několik překladů (počet: 0 nebo více).
    <mrkd> Je volitelný tag, který se používá k označení se cílového slovo v původní frázi a překladu
10. <co> Označuje text redakčního komentáře, který objasňuje smysl nebo kontext (zobrazeno v jiné barvě programem, obvykle šedá).
    Tip (indexování): komentáře jsou obvykle indexovány.
11. <sr> Je sekce věnovaná sémantickým vztahům k jiným slovům, jako jsou synonyma, holonyma, hyperonyma apod (XML 5). Využívá <kref> k adresování ostatních slov s dodatečnými atributy tagů.
    <sr> část vypadá takto:

```
<sr>
<kref type="syn" kcmt="obsolete">game</kref>
<kref type="hol" kcmt="partly">play</kref>
</sr>
```

XML 5 příklad sémantických vztahů [8]

Možné type honoty:

- syn a ant - synonyma a antonyma;
- hpr a hpn - hyperonyma a hyponyma (začleňují toponyma);
- par a spv - paronyma a pravopisné varianty;
- mer a hol - meronyma a holonyma;
- ent - vyplývání, speciální sloveso kategorie: v. Y je spojeno X, pokud používáte X, musíte použít Y, může být také použito pro podstatná jména:

podstatné jméno s sebou nese nějakou jinou věc (dělá "zločin" s sebou nese, že je "zločinec");

- `rel` - označuje význam (např., „hezké“ a „ošklivé“, jsou důležité pro „vzhled“);

`kcmt` atribut se používá k určení, do jaké míry je toto slovo souvisí s naším nebo specifikovat další informace o slově (jako je pohlaví nebo případ)

12. `<etm>` etymologické informace o slově.
13. `<di>` Označuje část `<def>` textu, které by neměly být indexovány. Mohly by být použity pouze v `<def>` a některé její děti: `<co>`, `<ex>`, `<etm>`, `<phr>`.
14. `<categ>` tag je poněkud ekvivalent kategorií Wikipedii (např. „výslech“ by mohlo mít kategorii „Středověká historie“)

3.2. Úpravy pro uživatelskou rozšiřitelnost

XDXF má hodně možností, avšak pro můj program jsem nevyužil všechny a nedodržuji některé zásady stanovené na začátku kapitoly. Slovník si uživatel může uložit do umístění a pojmenovat dle libosti, avšak doporučuji formát „z-do“ (např. `ENG-CZE`). Abych umožnil uživatelů reagovat na přesnost překladů, tak jsem přidal definicím dva atributy a využil třetí.

Jelikož tento typ formátování XML byl pouze definován v DTD, tak jsem jej prvně musel převést do XML schématu pomocí dostupné literatury[1]. Tento převod se nevyhnul potížím, protože XSD mnohem přesněji popisuje strukturu XML dokumentu.

```
<!ELEMENT def
((gr?,tr*,rref*,def+,ex*,sr?,etm?,categ*)|(gr?,tr*,(#PCDATA,dtrn|kref|rre
f|iref|abbr|c|ex|co|i|b|gr)*,sr?,etm?,categ*)) >
```

XML 6 Úprava v DTD [8]

V této ukázce kódu (XML 6) jsem musel odstranit element `<sr>`, protože překladač nevěděl, který element má použít (ambiguous).

contributors

Tento atribut (XML 7) poukazuje na to, kolik uživatelů hlasovalo pro určitý překlad. Tudíž čím více hlasů, tím je větší pravděpodobnost pro to, že je překlad doopravdy přesný.

```
<xs:attribute name="contributors">
  <xs:simpleType >
    <xs:restriction base="xs:unsignedShort">
      <xs:minInclusive value="0"/>
    </xs:restriction>
  </xs:simpleType>
</xs:attribute>
```

XML 7 Atribut user_freq v XSD

freq

Tento atribut je již implementován v originálním standardu. Já jej zde používám jako již vypočítaný

aritmetický průměr všech uživatelských hlasování o přesnosti překladu (XML 8). Atribut může nabývat hodnot od 0 do 10.

- 0: překlad je velice nepřesný.
- 10: překlad odpovídá nejvíce realitě.

```
<xs:attribute name="freq">
  <xs:simpleType >
    <xs:restriction base="xs:double">
      <xs:minInclusive value="0"/>
      <xs:maxInclusive value="10"/>
    </xs:restriction>
  </xs:simpleType>
</xs:attribute>
```

XML 8 Atribut user_freq v XSD

userFreq
Toto je uživatelský názor o přesnosti překladu (XML 9). Může nabývat hodnot od -1 do 10.

- -1: překlad je nesprávný a chci, aby byl odstraněn z oficiálního slovníku. Použiji jej tehdy, když v oficiálním slovníku naleznu překlad, který je špatný.
- 0: překlad je velice nepřesný.
- 10: překlad odpovídá nejvíce realitě.

```
<xs:attribute name="user_freq" >
  <xs:simpleType >
    <xs:restriction base="xs:int">
       <xs:minInclusive value="-1"/>
       <xs:maxInclusive value="10"/>
     </xs:restriction>
   </xs:simpleType>
 </xs:attribute>
```

XML 9 Atribut user_freq v XSD

Shrnutí

Formát slovníku XDXF je výhodný v tom, že je již delší dobu používán. V případě, že jsou v něm používány pouze mnou nadefinované prvky, je možné vypůjčovat si slovníky i z jiných programů. Opačným způsobem to nelze kvůli tomu, že byly přidaný atributy pro ty uživatele, kteří hodnotí přesnost překladu. V programu nejsou využívány všechny možností, které XDXF formát nabízí. Ty by bylo možno dodat, anebo odstranit teprve později, po uživatelské zpětné vazbě.

4. Aktualizace slovníku

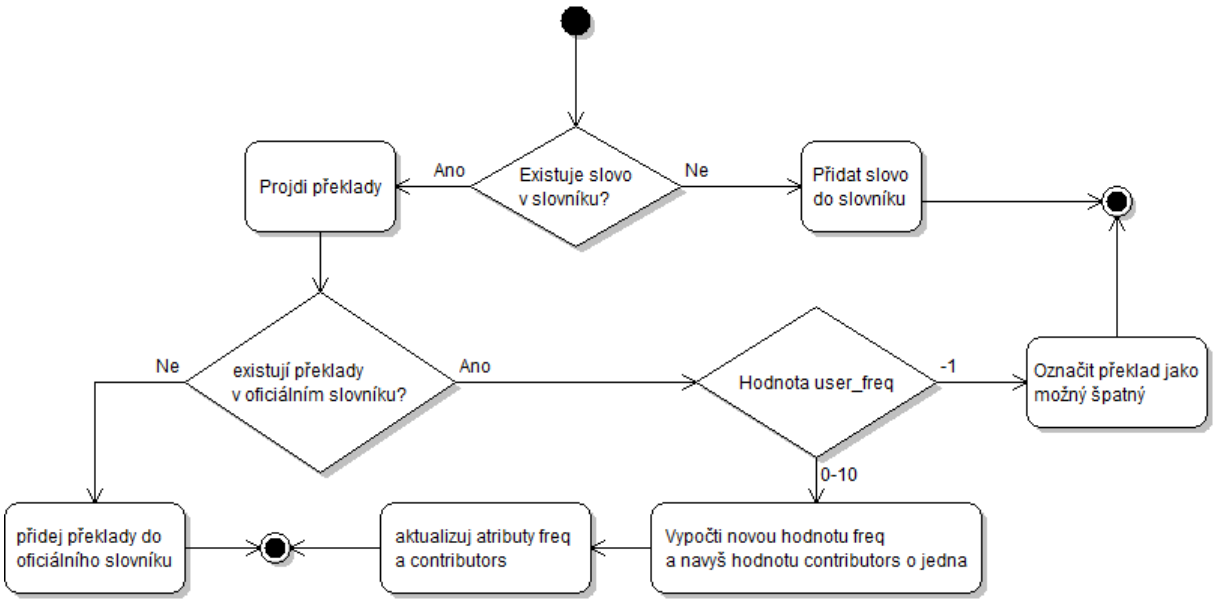
Každý uživatel má možnost vytvořit si vlastní slovník. Nad daty slovníku uživatelského je možno provádět CRUD operace, slovník oficiální je pouze pro čtení. Tento slovník bude posléze možné sdílet s ostatními uživateli nahráním na server přes webové rozhraní. Po každém nahrání na server se tento uživatelský slovník strojově spojí s oficiálním. Tato aktualizace není implementovaná.

- Slova dosud nejsou neobsažena v oficiálním slovníku, budou do něj přidána.
- Neznámé překlady budou rovněž přidány.
- U překladu, který již existuje v oficiálním slovníku, se načte uživatelské hodnocení. Vypočte se nový aritmetický průměr hodnoty překladu:

$$freq_{new} = \frac{(freq_{old} * contributors) + user\_freq}{(contributors + 1)}$$

Výše uvedeným výsledným aritmetickým průměrem se hodnoty v oficiálním slovníku aktualizují.

Pokud hodnota `user_freq` činí -1, je touto hodnotou překlad označen jako sice možný, nicméně špatný. Překlad bude z oficiálního slovníku zcela vymazán, v případě, že k výše uvedenému výsledku dojde ten počet uživatelů, který je určen správcem slovníku (Obrázek 8).



Obrázek 8 Vložení slova, diagram aktivit

4.1. Další slovníky

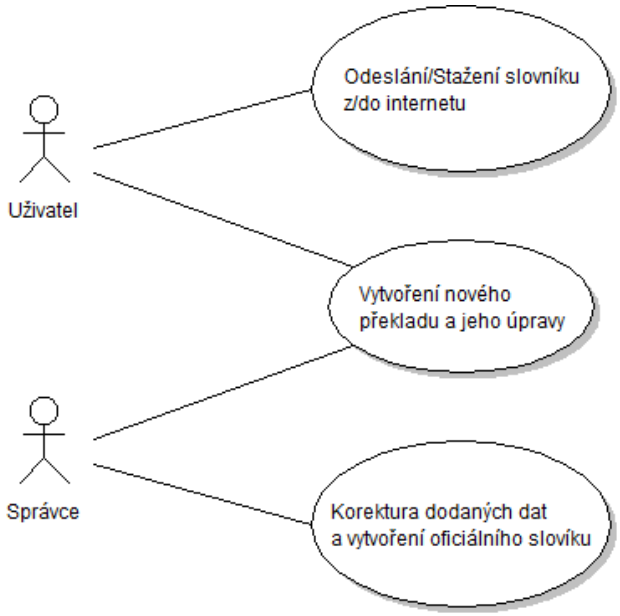
Zdroj dat, který používám pro oficiální slovník, čerpám z „GNU/FDL Anglicko-Český slovník“, podobně jako to činí SlovniQ (viz kapitola 2.4.). V případě vzniku nového slovníku, pro překlad dalších jazyků, by bylo zapotřebí vytvořit novou databázi o objemu asi 5000 slov.

4.2. Kontrola kvality

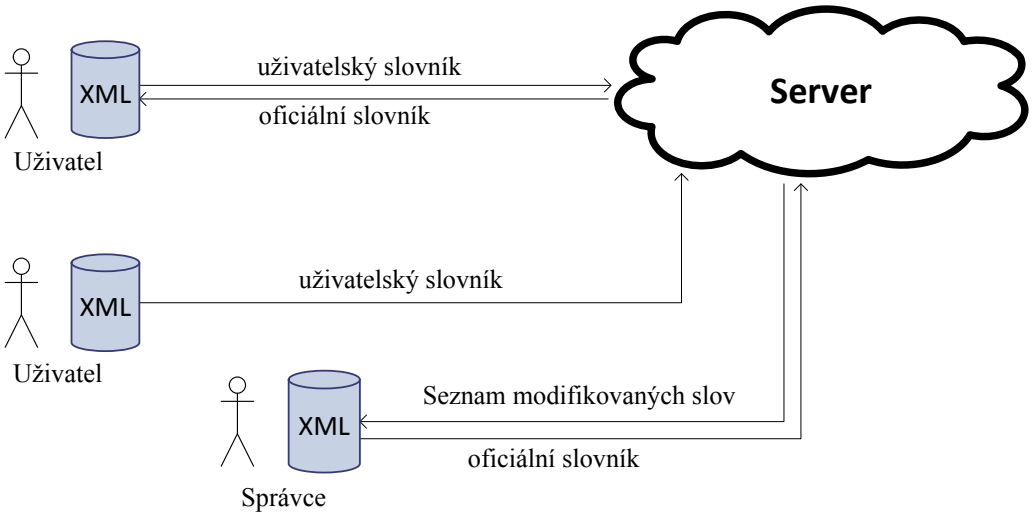
U strojového spojení slovníku však vzniká možnost výskytu chyb. Proto by bylo vhodné, aby zde existovala osoba Správce, která bude provádět jejich korekturu (Obrázek 9). Správce by pravidelně prošel pouze upravená/nová slova a zkontroloval je (Obrázek 10). Po takové kontrole správce vytvoří novější verzi oficiálního slovníku a poskytne ho uživatelům ke stažení.

Na serveru bude ke stažení k dispozici:

- Strojově spojená verze slovníku nezkontrolovaná pověřenou osobou; možné nesrovnalosti.
- Ověřený oficiální slovník.



Obrázek 9 Práva



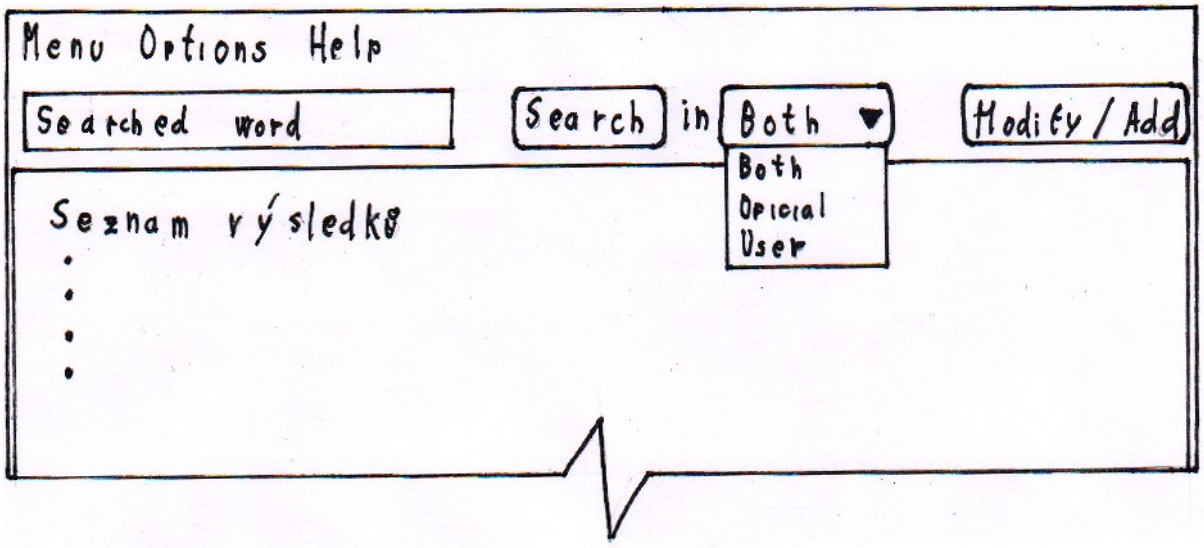
Obrázek 10 Aktualizace slovníku

5. Uživatelské rozhraní

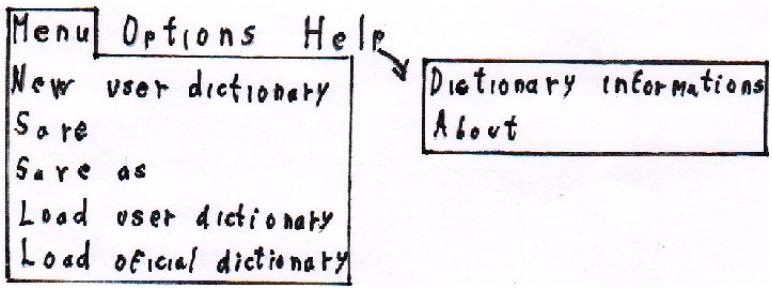
K tomu, aby uživatel mohl se slovníkem pracovat jednoduše a efektivně, je nezbytné mít k dispozici dobře přehledné grafické rozhraní. Dle doporučení vedoucího jsem všechna GUI navrhl a nakreslil ručně.

5.1. Hlavní okno

Jednoduché GUI, které umožní uživateli vyhledat slovo dle jeho výběru a to buď v slovníku uživatelském, oficiálním, nebo v obou zároveň (Obrázek 11). Pokud stejný překlad existuje v oficiálním i uživatelském slovníku, uživatelský má přednost a oficiální překlad se nezobrazí. Když napíšeme hledané slovo, můžeme vyhledávat kliknutím na `Search`, nebo zmáčknutím klávesy `Enter`. Pokud chceme upravit již existující překlad z oficiálního slovníku, musíme prvně zaškrtnout `Copy Oficial into User.` Toto okno obsahuje standartní menu (Obrázek 12).



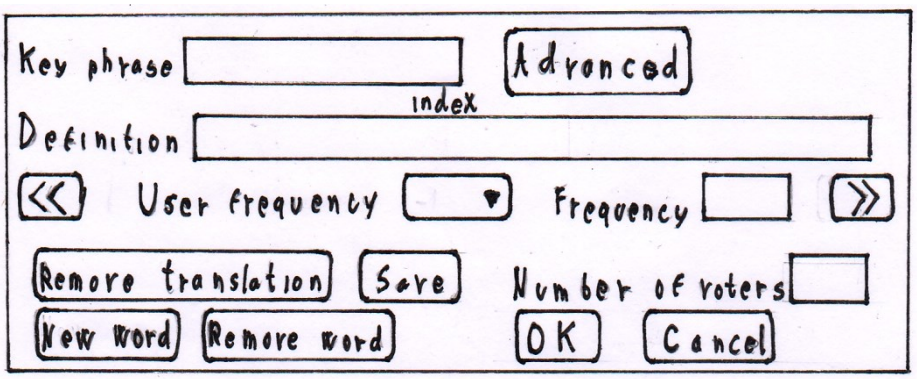
Obrázek 11 Hlavní obrazovka



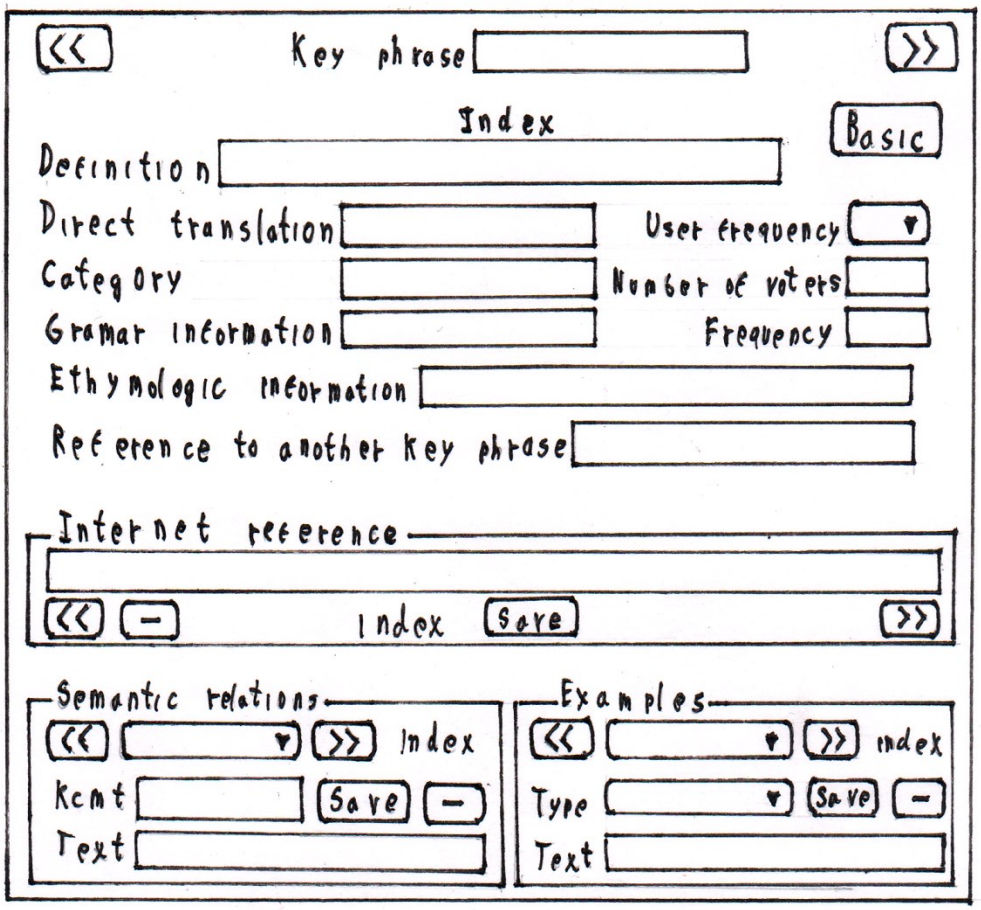
Obrázek 12 Menu

5.2. Modifikace záznamů

V tomto okně je možné provádět CRUD operace nad překlady. Každý překlad potřebuje alespoň jednu definici. Protože formulář tvorby překladu je složitý, tak jsem jej rozdělil na dvě části. Základní úprava slov (Obrázek 13), kde je pouze klíčové slovo a jeho definice. Pokročilá úprava slov nabízí všechny možnosti překladu (Obrázek 14).



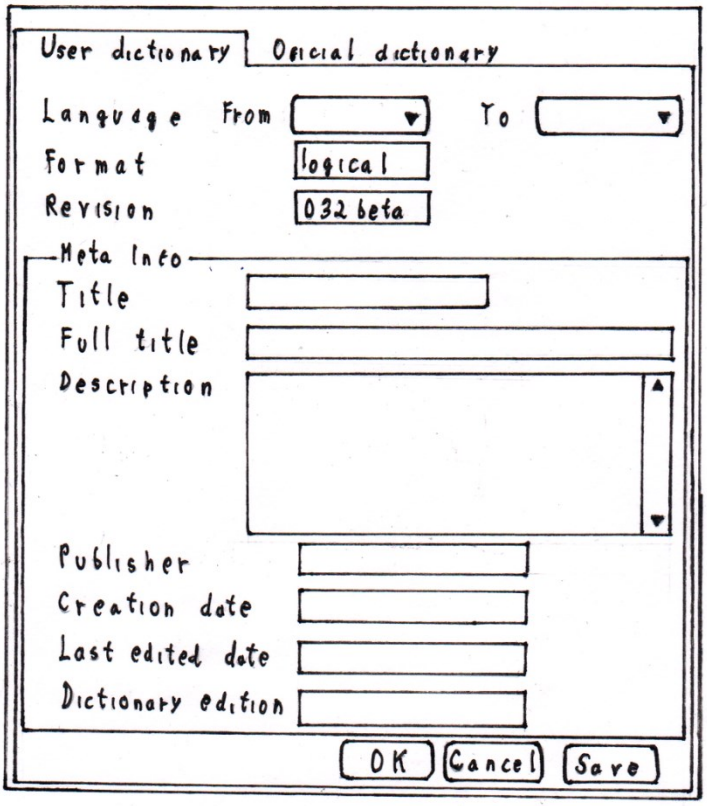
Obrázek 13 Základní úprava překladu



Obrázek 14 Pokročilá úprava překladu

5.3. Informace o slovníku

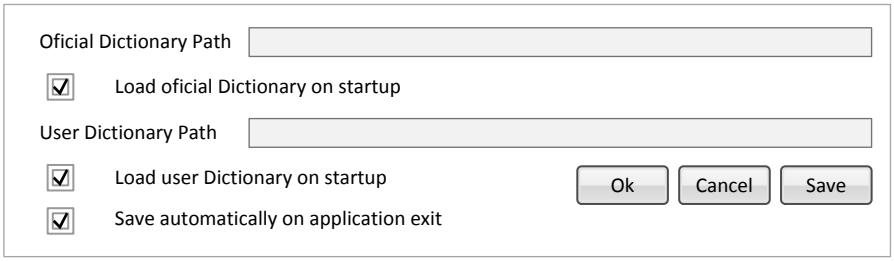
Zde si uživatel může nastavit informace popisující slovník (Obrázek 15). U oficiálního slovníku nemohu měnit žádná data.



Obrázek 15 Informace o slovníku

5.4. Nastavení

Uživatel se může rozhodnout, jestli chce současně otevřený slovník (uživatelský či oficiální) automaticky načítat při startu programu (Obrázek 16). A vybrat jak se bude program chovat při zavření. Automaticky uložit data, nebo upozornit že se data ztratí při uzavření. Po doporučení vedoucího, jsem nevytvořil jeden návrh ručně.

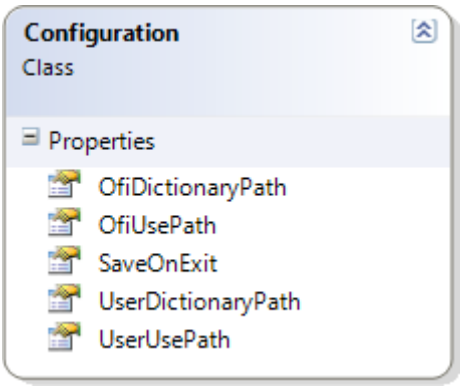


Obrázek 16 Nastavení

6. Aplikační rozhraní

6.1. Konfigurace

Uživatelské volby ukládám do odděleného souboru `config.cfg`. Tento soubor se načte při spuštění programu do instance třídy (Obrázek 17).



Obrázek 17 Třída Konfigurace

Konfiguraci načítám pomocí DOM (Document Object Model) a je uložena v paměti programu dokud jej nezavřeme (Zdrojový kód 6). V programu existuje pouze jedna instance načtené konfigurace. Při změně se konfigurace uloží do souboru na disku.

```
public static Data.Configuration LoadConfig()
{
    Data.Configuration cfg = new Data.Configuration();
    XmlDocument lDoc = new XmlDocument();
    lDoc.Load("config.cfg");
    XmlNode lRoot = lDoc.DocumentElement;
    XmlAttributeCollection RootAttrColl = lRoot.Attributes;
    for (int i = 0; i < RootAttrColl.Count; i++)
    {
        switch (RootAttrColl[i].Name)
        {
            .
            .
            .
        }
    }
     return cfg;
}
```

Zdrojový kód 6 Konfigurace, načtení

Při startu programu se automaticky vyhledá soubor s uloženou konfigurací, když neexistuje, tak se vytvoří při změně.

6.2. Vyhledávání slova

XDXF umožnuje, aby každý překlad měl několik klíčových slov. Kvůli rychlosti vyhledávání a pravděpodobnosti že takovýchto slov bude málo, využívám u každého slova pouze jedno klíčové slovo.

```
Modified = new Data.Record();
if (comboBoxIn.Text.Equals("User") || comboBoxIn.Text.Equals("Both"))
    {
    if (UserDictionary.Lexicon.ContainsKey(txtSearch.Text))
        {
        rtxtMain.SelectionFont = new Font(rtxtMain.SelectionFont,
FontStyle.Italic);
        rtxtMain.AppendText("User translation:\r\n");
        List<Data.Definition> value = new List<Data.Definition>();
        value = UserDictionary.Lexicon[txtSearch.Text];
        Modified.K = txtSearch.Text;
        foreach (Data.Definition pom in value)
        {
        Data.Definition defi = new Data.Definition();
        defi.Def = pom.Def;
        defi.UserFreq = pom.UserFreq;
        defi.Contibutors = pom.Contibutors;
...
```

Zdrojový kód 7 Vyhledávání

Při vyhledávání se prvně vytvoří nová instance pro modifikované slovo. Poté se zjistí, kde chci vyhledávat, zkontroluje se, jestli existuje ve slovníku a až nakonec se provádí samotné vyhledávání (Zdrojový kód 7). Zapisují se nalezená data do prvku `RichTextBox`, toto dovoluje formátovat nalezené záznamy lépe než obyčejný `TextBox`. Automaticky se ukládají vyhledaná data z uživatelského slovníku do proměnné `Modified`, jestliže chceme také modifikovat data z oficiálního slovníku, tak musíme zaškrtnout `CheckBox` označený `Copy Oficial into User`, takto modifikované záznamy se uloží do uživatelského slovníku. Při zjištění, že ve slovnících jsou stejné překlady, tak uživatelský slovník má větší prioritu a zobrazí se pouze jeho překlad.

Kdyby se modifikovalo vyhledané slovo přímo, tedy skrze reference na klíč a definice, vznikl by problém, při potvrzování změn. Tudíž kdyby se změnila například definice a poté bychom se rozhodli změny vrátit, bylo by to obtížné. Proto při vkládání dat do proměnné `Modified se` vytváří nová instance definice pro každý záznam v seznamu definic. Do této instance se vkládají hodnoty po jedné, aby se ukládala hodnota a ne reference.

6.3. Úprava slova

Upravované slovo je to, které jsme vyhledali, nebo pokud nemáme nic vyhledané, tak vytváříme nové slovo. Jak jsem se již zmínil, abych nezahltil uživatele informacemi, rozdělil jsem úpravu překladu na složitější a jednoduší.

Při kliknutí na modifikaci slova se spustí obě okna (formy)(Zdrojový kód 8), přičemž spolu komunikují. Kdybychom měli slovo, které má čtyři překlady a modifikovali bychom překlad s indexem dva a přepnuli se do rozšířené modifikace slova, tak by stále byl zobrazen index dva.

```
advanced = new AdvancedAddModify();
basic = new BasicAddModify();
basic.Show();
```

Zdrojový kód 8 Inicializace modifikování slova

Pro navigaci mezi překlady používám indexy, první index představuje konkrétní položku a druhý počet překladů ve slově, například 2/4. Pokud jsme na indexu 0, tak můžeme přidávat nový překlad kliknutím na tlačítko Save. Přidaný překlad musí mít vyplněnou definici a uživatelské hodnocení překladu. Když jsme na jiném indexu, tak upravujeme existující překlad.

Stejný způsob navigace používám i pro příklady, internetové odkazy a sémantické vztahy.

Potvrzení modifikace slova

Při kliknutí na tlačítko OK se uzavře okno a podle úprav které jsme provedli, se slovo ve slovníku aktualizuje. Uložením slova se překlady seřadí podle uživatelského hodnocení sestupně (Zdrojový kód 9).

```
Main.Modified.Def = Main.Modified.Def.OrderByDescending(x =>
x.UserFreq).ToList();
```

Zdrojový kód 9 Seřazení překladů

Nastaví se hodnota save na true, tímto způsobem program kontroluje, jestli byla provedena změna ve slovníku. Musí se také kontrolovat, jestli již ve slovníku existuje klíčové slovo (Zdrojový kód 10).

```
if (!Main.UserDictionary.Lexicon.ContainsKey(Main.Modified.K))
{
    Main.save = true;
    if (OldKey!=Main.Modified.K)
    {
        if (OldKey != "")Main.UserDictionary.Lexicon.Remove(OldKey);
        Main.UserDictionary.Lexicon.Add(Main.Modified.K,
Main.Modified.Def);
        if (OldKey != "") Main.UsernamesCollection.Remove(OldKey);
        Main.UsernamesCollection.Add(Main.Modified.K);
    }
    else
    {
        Main.UserDictionary.Lexicon.Add(Main.Modified.K,
Main.Modified.Def);
        Main.UsernamesCollection.Add(Main.Modified.K);
    }
}
```

Zdrojový kód 10 Slovník obsahuje klíč

Když klíčové slovo neexistuje, tak také se zkontroluje, jestli jsme jej nepozměnili. Když ano, tak se odstraní starý klič ze slovníku a vloží se nový. V ostatních případech se pouze přidá nové slovo do slovníku. Také se aktualizuje slova, které nám slovník nabízí na doplnění (Zdrojový kód 11).

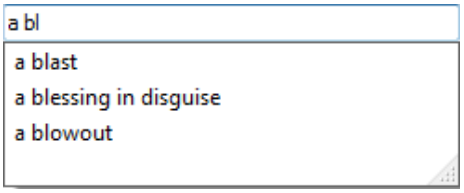
```
else
{
    Main.save = true;
    if (OldKey.Equals(Main.Modified.K))
    {
        Main.UserDictionary.Lexicon[Main.Modified.K] = Main.Modified.Def;
    }
    else
    {
        Main.UserDictionary.Lexicon.Remove(OldKey);
        Main.UserDictionary.Lexicon.Add(Main.Modified.K,
Main.Modified.Def);
        Main.UsernamesCollection.Remove(OldKey);
        Main.UsernamesCollection.Add(Main.Modified.K);
    }
}
```

Zdrojový kód 11 Slovník neobsahuje klíč

Podobně se pracuje při tom, že slovník obsahuje klíč. Zkontrolujeme, jestli nebyl pozměněn klič modifikovaného slova. Pokud ne, tak se aktualizují novými data. Když ano, tak se odstraní starý klíč a je nahrazen novým.

6.4. Automatické doplňování

Při psaní vyhledávaného slova se zobrazují podobná slova (Obrázek 18). Kliknutím na toto slovo se vloží do `TextBoxu`. Při načtení slovníků se vyplní seznam slov pro automatické doplňování.



Obrázek 18 Automatické doplnění

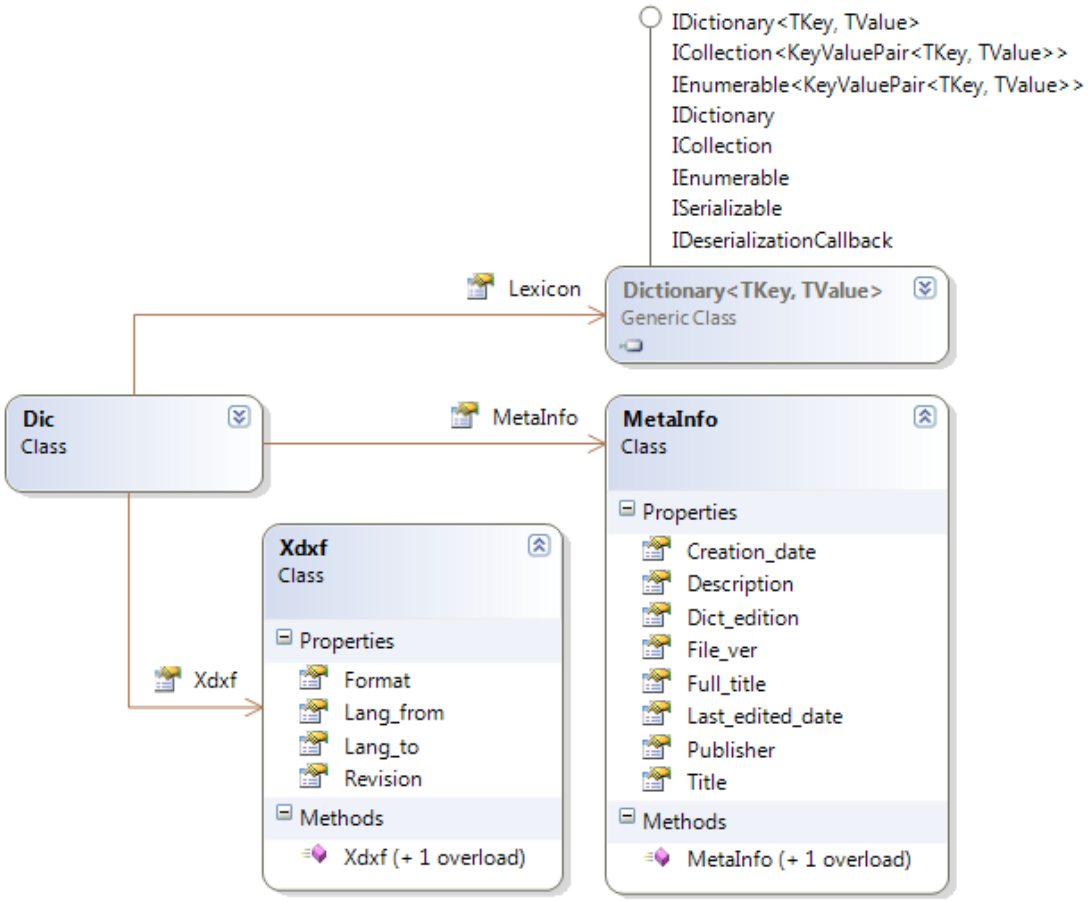
Jelikož zdroj pro automatické doplnění může být pouze pole řetězců, bylo potřeba toto omezení obejít. Proto se naplní `List` a tím se poté naplní pole řetězců a až nakonec se vloží jako zdroj dat (Zdrojový kód 12).

```
public static List<string> OfinamesCollection = new List<string>();
public static List<string> UsernamesCollection = new List<string>();
string[] oficialCollection = OfinamesCollection.ToArray();
string[] userCollection = UsernamesCollection.ToArray();
txtSearch.AutoCompleteCustomSource.AddRange(oficialCollection);
txtSearch.AutoCompleteCustomSource.AddRange(userCollection);
```

Zdrojový kód 12 Automatické doplňování, deklarace a naplnění

6.5. Slovník

V programu můžeme mít načteny dva slovníky, oficiální a uživatelský, každý pouze jednou. Každý slovník obsahuje hlavičku `XDXF`, `MetaInfo` a spárovaný seznam klíčových slov s definicemi (Obrázek 19). Po doporučení vedoucího se slovník načítá do paměti.



Obrázek 19 Třída představující slovník

Instance slovníků jsou vytvořeny ve `formu Main`. K těmto slovníkům přistupuji v celém programu, proto jsou `public` (Zdrojový kód 13).

```
public static Data.Dic OfiDictionary = new Data.Dic();
public static Data.Dic UserDictionary = new Data.Dic();
```

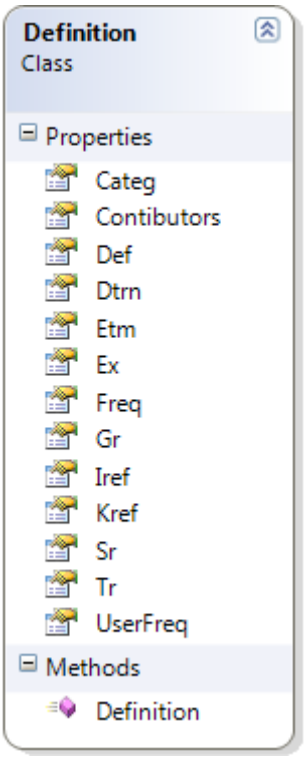
Zdrojový kód 13 Inicializace slovníků

Při spuštění programu se načte prvně konfigurační soubor, pokud máme zatrženo v nastavení, že se má slovník načíst, tak to udělá. Jestliže slovník neexistuje, program zobrazí varování.

Skrze funkci v `Menu/Load User Dictionary` lze pouze načíst uživatelský slovník, pokud se pokusíme takto načíst jakýkoli jiný slovník, program nás na to upozorní. Podobně to platí i pro oficiální slovník. Funkce pro načtení slovníků z XML obstarává tuto funkci. To znamená, když metoda zjistí při načítaní hlavičky, že je použita nesprávně, ukončí načítání.

6.6. Překlad

Každé slovo musí mít alespoň jeden překlad. Samotné slovo je spárování klíčového slova s definicí popřípadě definicemi. Ve slovníku je klíčové slovo unikátní, může se vyskytovat pouze jednou. Každý překlad musí mít jednu definici (Obrázek 20).



Obrázek 20 Definice překladu

Tato metoda je představuje data načtená z XML. Každý její prvek představuje určitou část překladu, z toho pouze odkazy na internetové stránky, příklady a sémantické vztahy se mohou vyskytovat vícekrát. XDXF sice dovoluje, aby některé další prvky se vyskytovaly v překladu také několikrát, avšak kvůli zjednodušení modifikace a přidávání slov jsem je zjednodušil. Při dalším rozvoji se mohou přidat další prvky překladu, které jsou definované v XDXF.

6.7. Validace XML

Před načtením slovníku z XML souboru do paměti se soubor validuje. Zkontroluje se, jestli rozložení XML se shoduje s XML schématem definovaném v XSD. Když objeví chybu, dokument se nenačte (Zdrojový kód 14). Tuto funkci jsem přidal dodatečně, protože veškerá tvorba a úprava slovníku se provádí pomocí programu. Tedy když je chyba v XML souboru, je také chyba v programu. Kontrolu neprovádí ideálně, pro každou chybu se otevře vlastní okno.

```
bool errors = false;
string err = "";
doc.Validate(schemas, (o, e) =>
{
    err += e.Message + "/n/r";
    errors = true;
});
if (errors)
{
    DialogResult result = MessageBox.Show("Errors:\n\r" + err,"Errors",
                                    MessageBoxButtons.OK);
     return true;
}
else return false;
```

Zdrojový kód 14 Validace

Shrnutí

Program obsahuje 6 382 řádků kódu, 8 tříd a 6 oken. Hojně byla využívána knihovna `System.Xml` a sporadicky i `System.Linq`. Program nebyl testován jiným programátorem (development testing), avšak bylo provedeno akceptační testování prostřednictvím uživatelů. Jedné skupině testerů byl nabídnut přístup k celému zdrojovému kódu (white box)[9][10]. Druhá skupina testovala pouze program, aniž by věděla, jak program funguje (black box)[9][10]. Díky těmto uživatelům jsem tak získal užitečnou zpětnou vazbu. Tímto způsobem jsem dovršil své úsilí vytvořit bezchybný program. Uvědomuji si však, že dalším užíváním bude možno vyřešit problémy, které se posléze pravděpodobně ještě objeví.

7. Budoucí rozšíření

Jako všechny programy je možno i tento vyvíjet. Při vytváření tohoto slovníku jsem se snažil potřeby uživatelů předvídat. I když nelze vyhovět všem, lze předpokládat, že doposud implementovaná nosná kostra slovníku, poslouží jako solidní základ, na kterém se posléze bude moci dále stavět. K úpravám programu lze přistoupit po jeho dalším testování.

7.1. Data překladu

Jako samotný formát XDXF nabízí velkou možnost v oblasti, co všechno může samotný překlad obsahovat. Co uživatelská komunita bude očekávat, a používat v překladech jsem vytvořil podle mé představy. Například: průměrný uživatel nebude potřebovat znát etymologický význam slova, nebo bude chtít přidat přehrávání výslovnosti slova a podobně.

7.2. Snadné přepínání mezi slovníky

Při současném užití více slovníků může docházet k určité prodlevě při jejich přepínání. Při spárování slovníků (Oficial, User) a uložení tohoto páru do konfiguračního souboru vznikne jednoduchý způsob jak mezi různými páry přepínat.

7.3. Webové rozhraní

Jak bylo již v předchozích kapitolách zmíněno, k využití slovníků v plném rozsahu je nutné externí úložiště dat, kde budou slovníky shromažďovány a udržovány. Tento server bude přístupný přes webové rozhraní. Bude muset být dostatečně velký a výkonný, aby byl schopen mít otevřeno více velkých slovníku současně. Při dnešních technologiích jsou servery bez větších problémů schopny dostát těmto požadavkům. Je totiž nutné brát do úvahy efektivitu zpracování dat, poněvadž zatím není zřejmé, do jaké velikosti se slovník časem rozroste.

Webové stránky budou potřebovat správce (administrátora), který je bude udržovat. A specialisty, kteří budou mít na starosti odlišné jazyky pro tvorbu oficiálních slovníků.

7.4. Změna XML úložiště za SQL

V případě nutnosti by bylo možné rozšířit program tímto způsobem. Tato změna představuje pouze možnost, kterou se může tento program rozvíjet, nemusí se uskutečnit, avšak ji pro úplnost uvádím.

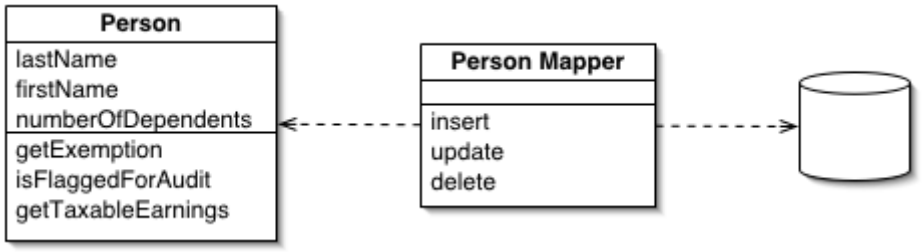
Při tvorbě programu jsem se inspiroval návrhovým vzorem Data Mapper: Vrstva mapperů, které přesouvají data mezi úložištěm a objekty přičemž jsou k sobě navzájem a mapperem nezávislé (Obrázek 21).

Tímto způsobem, pokud bychom nezměnili způsob, kterými pracujeme s daty, by bylo možně provést změnu pouze přepsáním třídy `ControlServices` a jejich metod. Hlavním rozdílem by bylo, že by

vznikla potřeba internetového připojení. Z toho také plyne, že vytvoření internetové verze slovníku by bylo snadno implementovatelné.

Možnosti uložení uživatelského slovníku:

- V lokální SQL databázi.
- V XML, bylo by složitější převádění dat mezi oficiálním SQL a uživatelským XML při spojování slovníků.



Obrázek 21 Data Mapper[1]

[1] FOWLER, Martin. *P of EAA* [online]. 2000 [cit. 2014-04-25]. Dostupné z: http://martinfowler.com/eaaCatalog/dataMapper.html

8. Závěr

Cíl zadání byl splněn: Slovník byl navržen a implementován. Bylo tak učiněno s ohledem na existující řešení i standardy tagů XML pro slovníky. Implementovaný prototyp byl shledán funkčním, neboť s uživatelskými dobrovolníky byl rovněž úspěšně proveden experimentální test funkčnosti a rozhraní slovníku. Celý návrh je formálně zdokumentován. Uživatelé jej mohou dále rozvíjet.

Dovolil jsem si připojit ještě několik osobních vyjádření:

Tento slovník jsem se snažil vytvořit co nejlépe, a skutečně jsem pro to vynaložil mnoho úsilí a času. Musím přiznat, že v průběhu práce se mi vynořovaly neustále další a další nápady pro inovaci svého slovníku. Tím bych se však jako vývojář dostal do začarovaného kruhu; kromě toho mě omezoval čas. I když by bylo možné při řešení zadání zvolit různé cesty, jsem přesvědčen, že způsob, který jsem zvolil já, je z hlediska první verze nejvhodnější. Při vývoji tohoto projektu mě velmi potěšilo, že jsem si mohl vyzkoušet, co všechno tvorba takového slovníku obnáší. Když jsem se potýkal s problémy, které se při vývoji vyskytly, získal jsem mnoho poučných zkušeností, které jsou pro mě inspirací k dalšímu rozvoji slovníku.

Jsem rád, že se mi podařilo vytvořit funkční prototyp slovníku přesně dle specifikací zadání vedoucího mé práce.

9. Reference

[1] MLÝNKOVÁ, Irena, POKORNÝ, Jaroslav. *XML technologie: principy a aplikace v praxi*. 1. vyd. Praha: Grada, 2008, 267 s. Průvodce (Grada). ISBN 978-80-247-2725-7.

[2] VIRIUS, Miroslav. *C# 2010: hotová řešení*. 1. vyd. Brno: Computer Press, 2012, 424 s. K okamžitému použití (Computer Press). ISBN 978-80-251-3730-7.

[3] *Atlantida Multilingual Dictionary* [online]. 2006 [cit. 2014-03-31]. Dostupné z: http://atla.revdanica.com/en/

[4] GOOGLE. *Google translate* [online]. 2007 [cit. 2014-03-31]. Dostupné z: https://support.google.com/translate/#

[5] What is reCAPTCHA. *ReCaptcha* [online]. 2014 [cit. 2014-03-31]. Dostupné z: http://www.google.com/recaptcha/learnmore

[6] *Duolingo* [online]. 2013 [cit. 2014-03-31]. Dostupné z: https://www.duolingo.com/

[7] NĚMEC, Jiří. *SlovniQ* [online]. 2010 [cit. 2014-03-31]. Dostupné z: http://code.google.com/p/slovniq/

[8] XDXF Description. *Github* [online]. 2013 [cit. 2014-04-01]. Dostupné z: https://github.com/soshial/xdxf_makedict/blob/master/format_standard/xdxf_description.md

[9] PATTON, R. Testování softwaru. SAMS, Computer Press, Praha, 2002, ISBN 80-7226-636-5

[10] PALETA, Petr. *Co programátory ve škole neučí: aneb Softwarové inženýrství v reálné praxi*. Vyd. 1. Brno: Computer Press, 2003, 337 s. ISBN 80-251-0073-1.

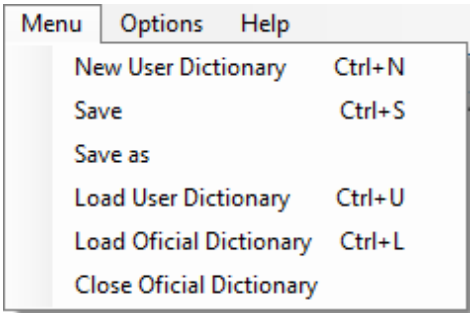
Příloha A. Obsah přiloženého DVD

- Text bakalářské práce
- Slovník a jeho zdrojový kód
- Data slovníku
- XSD formátování XDXF

Příloha B. Uživatelský manuál

Při prvním spuštění programu nejsou načteny žádné slovníky, tudíž uživatel musí načíst slovníky manuálně.

- Načtení uživatelského slovníku: kliknutím na *Menu/Load User Dicitionary* nebo *Ctrl+U* (Obrázek 22)
- Načtení oficiálního slovníku: kliknutím na *Menu/Load Oficial Dicitionary* nebo *Ctrl+L*



Obrázek 22 Menu

Po načtení slovníku si program vyžádá všechna klíčová slova a nabídne je pak při psaní uživateli (Obrázek 23) na doplnění (autocomplete). Je možné si vybírat, v kterém slovníku chceme vyhledávat (v obou slovnících; v slovníku uživatelském; v slovníku oficiálním).

Obrázek 23 autocomplete

Možnosti

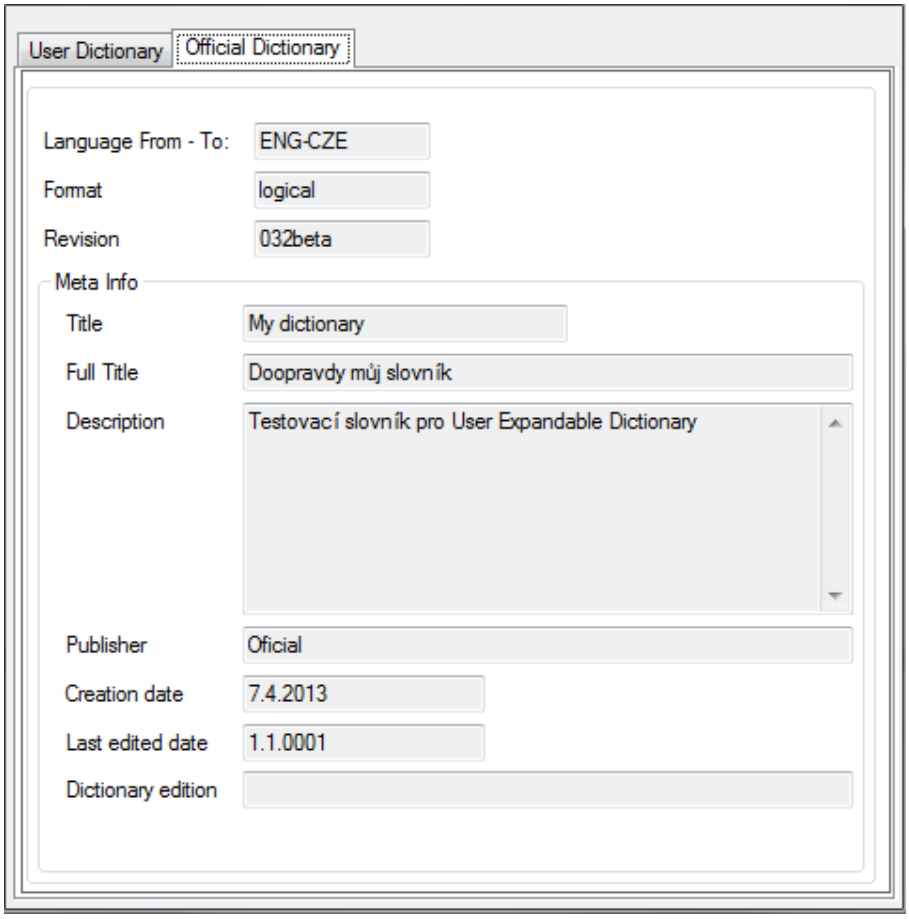
Na výběr jsou možnosti automatického načítaní naposledy načtených slovníků. Zobrazena je i cesta ke slovníku. Lze také vybrat jak se bude program chovat po svém uzavření:

- Automaticky uloží změněná data.
- Zobrazí varování, že v případě zavření budou veškeré změny ztraceny.

Příklad: Při zobrazování výsledků se zobrazí před překladem slova `„3 | 5/1“.`

- `3` je uživatelské hodnocení.
- `5` oficiální.
- `1` počet lidí, kteří přispěli.

Informace o slovníku



Obrázek 24 Dictionary information

Zde jsou zobrazeny informace popisující slovník (Obrázek 24). U oficiálního slovníku není možno měnit data. Uživatelský slovník má možnost změny dat až na několik výjimek.

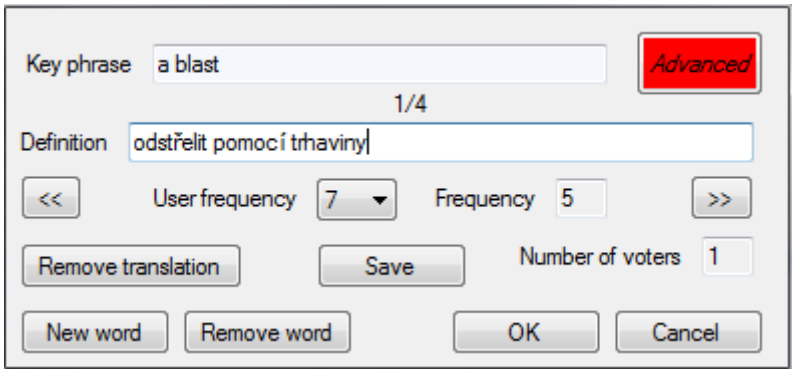
- Creation date: automaticky se vyplní při vytvoření nového slovníku.
- Last Edited Date: aktualizuje se pouze při uložení slovníku.
- Publisher: je vždy `User`, protože toto je uživatelský slovník.
- Format: může být `logical` nebo `visual`. Visual nepoužíváme, protože je zastaralý.
- Revision: verze revize XDXF. je vždy `032beta`.

Tato data je možné měnit.

- Language from: originální jazyk.
- Language to: překládaný jazyk.
- Title: jméno slovníku v angličtině.
- Full title: plné jméno slovníku v jakémkoli jazyce.
- Description: popis slovníku.
- Dictionary edition: edice slovníku.

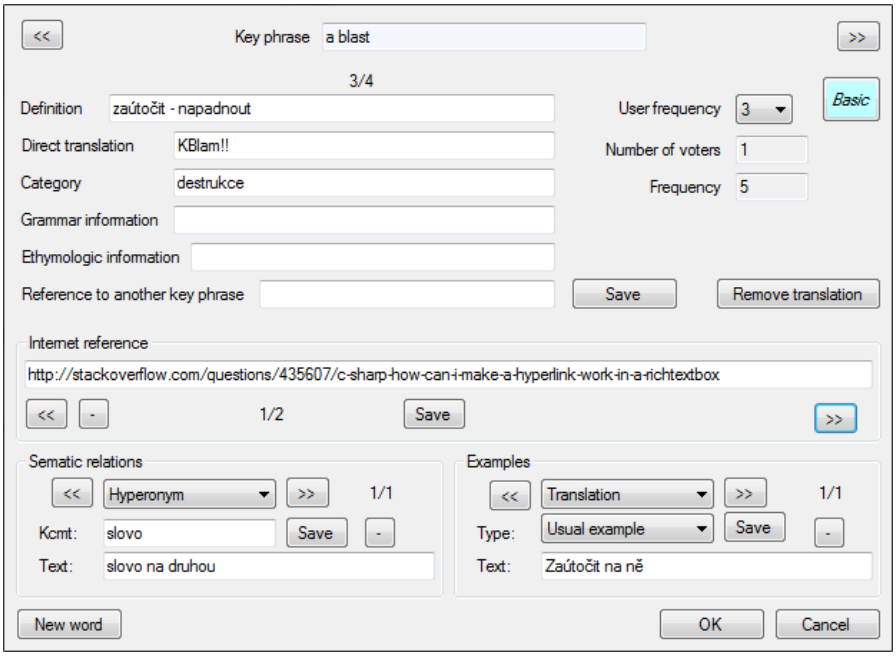
Přidávání a upravování slov

V případě, že chceme některé slovo upravit, musíme nejprve dotyčný výraz vyhledat, poté kliknout na `Modify/Add` (zobrazené okno viz Obrázek 25). Pokud chceme upravit slovo, které je v slovníku oficiálním, pak musíme nejprve zaškrtnout `Copy Oficial into User`. Pro provedení změny v překladu musíme kliknout na `Save`. Avšak pro uložení této změny do slovníku musíme ještě kliknout i na `OK`. Další překlad můžeme přidat, přejdeme-li šipkami na index 0. Na tento index se dostaneme, jsme-li na posledním indexu a klikneme >>, nebo kliknutím na << .To ovšem platí pouze tehdy, že jsme na prvním indexu.



Obrázek 25 Basic Add/Modify

Kliknutím na `Advanced` se dostaneme do rozšířené možnosti modifikování překladu (Obrázek 26).



Obrázek 26 Advanced Add/Modify

Zde navigace a další funkce pracují stejně jako v základní modifikaci překladů. Pro přidání jakékoli další kategorie (např. Example apod.) nesmíme být na indexu 0 překladu.